Aquarius Level 3 processing algorithm theoretical basis document, Part II. Implementation

Jonathan M. Lilly and Gary S. E. Lagerloef

February 27, 2009

Abstract

This document presents the theoretical basis of local polynomial fitting, its implementation, and some preliminary applications to producing Level 3 gridded files from the Aquarius sampling pattern.

1 Introduction

Smoothing scattered data is one of the main problems in statistics, and as such, a wide variety of methods exist. In oceanography, partly for historical reasons, optimal interpolation—also known as objective mapping or Gauss-Markov—tends to be the method of choice. After reviewing this and variety of other methods, we concluded that "local polynomial fitting" would be most appropriate for Aquarius. This is because the smoothing radius is explicitly controlled, and yet the degree of smoothness can be modified with a second parameter, namely the order of the fit. The reader is referred to our earlier document for more discussion of our logic here.

As the name implies, local polynomial fitting involves a least squares fit of the data to a polynomial. This fit is performed "locally" in vicinity of each grid point, by weighting nearby data with a decaying function. The order of the fit (constant, linear, quadratic, etc.) impacts the degree of smoothness of the resulting surface. The linear fit takes into account an estimate of the local gradient, and the quadratic fit also takes estimated curvature into account.

This document introduces the basic idea of local polynomial fitting, presents equations for the solution to scattered data over the (x, y) plane following Ruppert and Wand (1994) and Fan and Gijbels (1996), derives new equations for the solution to scattered data over the sphere, and presents some preliminary applications to output from the Aquarius simulator.

2 The basic idea

Let's say we have a surface, or field, $z(x, y)$ which is a function of two coordinates, which we can call distance east x and distance north y. In our case, of course, $z(x, y)$ is the sea surface salinity. The surface is sampled at Q data points $\tilde{z}_q$ at x-locations $\tilde{x}_q$ and y-locations $\tilde{y}_q$, with $q = 1, 2, \ldots, Q$. We arrange the data and the sample coordinates into vectors of length Q such as $\tilde{\mathbf{z}} = [\tilde{z}_1 \ \tilde{z}_2 \ \ldots \ \tilde{z}_Q]^T$ where "T" indicates the matrix transpose, and similarly for $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$. Based on this irregularly sampled data, we desire to obtain estimates of $z(x, y)$ on a regular grid. Here we use the notation "$\tilde{z}$" to indicate a (potentially noisy) observation of z, and the notation "$\widehat{z}$" to indicate a gridded estimate of z based on those observations.

Local polynomial fitting is based around a local Taylor series expansion. In the vicinity of a particular point (x_o, y_o), one may approximate the surface $z(x, y)$ by a Taylor series expansion,

$$z(x, y) = z(x_o, y_o) + (x - x_o) \frac{\partial z}{\partial x}(x_o, y_o) + (y - y_o) \frac{\partial z}{\partial y}(x_o, y_o) + \epsilon^{(2)}(x, y, x_o, y_o) \qquad (1)$$

where the last term $\epsilon^{(2)}$ is an error term arising from higher-order variations. Here, we have resolved terms up to first order—the linear terms—but we could have also included the quadratic terms, or alternatively, we could truncate the expansion with the zeroth order (constant) term.

In local polynomial fitting, one takes available data in the vicinity of a certain point (x_o, y_o), and uses this data to estimate not just the value of the surface $z(x_o, y_o)$ but also the values of its derivatives. For a zeroth-order fit, only the value of the surface is estimated; for a first-order fit, the value of the surface and of its gradient is estimated; and for a second-order fit, six quantities are estimated—the surface, its gradient, and three second-order derivatives. This estimation is accomplished though a *least squares fit*. The zeroth-order fit corresponds to finding the best *constant* value which matches the data in the vicinity of (x_o, y_o), the first-order fit corresponds to finding the best *plane*, and the second-order fit corresponds to finding the best *parabolic surface*.

A weighting function is necessary in order to localize the least-squares fit to the vicinity of the point (x_o, y_o). We use the decaying parabola

$$K_B(r) = \begin{cases} 1 - \frac{r^2}{B^2} & r/B < 1 \\ 0 & r/B \geq 1 \end{cases} \tag{2}$$

where r is the distance between two points. The function $K_B(r)$ decays from a maximum of unity at $r = 0$, and vanishes for $r \geq B$. The radius of the fit is thus controlled by the parameter B, which is known as the "bandwidth". There is a theoretical justification for using the parabolic function as opposed to, say, a Gaussian, as discussed in Fan and Gijbels (1996). The fact that the weighting function vanishes exactly outside radius B also makes the computation easier, as one does not have to keep track of small contributions from very large distances, as one would with a less compact function like a Gaussian.

Let's say we have Q data points at x-locations $\tilde{x}_q$ and y-locations $\tilde{y}_q$ observing values $\tilde{z}_q$, with $q = 1, 2, \ldots, Q$, and let the distance to the grid point be denoted by

$$r_q(\tilde{x}_q, \tilde{y}_q, x_o, y_o) \equiv \sqrt{(\tilde{x}_q - x_o)^2 + (\tilde{y}_q - y_o)^2}. \tag{3}$$

We use the notation "$\tilde{z}$" to indicate a (potentially noisy) observation of z, and the notation "$\widehat{z}$" to indicate an estimate of z based on those observations. To find the solution to the local first-order fit, one minimizes the weighted mean-square error

$$\sum_{q=1}^{Q} \left| \tilde{z}_q - \widehat{z}(x_o, y_o) - (\tilde{x}_q - x_o) \frac{\widehat{\partial z}}{\partial x}(x_o, y_o) - (\tilde{y}_q - y_o) \frac{\widehat{\partial z}}{\partial y}(x_o, y_o) \right|^2 K(r_q) \tag{4}$$

by varying the three quantities $\widehat{z}(x_o, y_o)$, $\frac{\widehat{\partial z}}{\partial x}(x_o, y_o)$, and $\frac{\widehat{\partial z}}{\partial y}(x_o, y_o)$. These correspond, respectively, to the estimated value of the surface $z(x, y)$, the estimated value of its x-derivative, and the estimated valued of the y-derivative, all evaluated at the grid point (x_o, y_o). The solution is known by linear least squares theory and is given later. Performing this least squared fit at a variety of different grid points (x_o, y_o), we obtain the desired solution on a regular grid.

The order of the fit, and the bandwidth B, constitute two degrees of adjustability of local polynomial fitting. It might seem that the different order fits would give the same answer for the estimated value of the surface $\widehat{z}(x_o, y_o)$, but this is not the case. Particularly when the data spacing is irregular, there can be a considerable improvement in the estimation of the surface when the higher-order terms are taken into account. When the data is noisy, as is usually the case, there is a tradeoff to increasing the order of the fit. As the fit order increases, the local structure of the surface is better represented, but more parameters need to be estimated. Thus a higher-order fit decreases *bias*, or modeling error, at the expense of *variance* or random error. A variety of methods exist for finding the "best" values of the bandwidth and of the fit order (Fan and Gijbels, 1996).

3 Matrix equations

We can write down a compact form for the local polynomial fit over the (x, y) plane, following Ruppert and Wand (1994) and Fan and Gijbels (1996). We denote the order of the polynomial fit by ρ, and consider fits of order $\rho = 0$ (constant), $\rho = 1$ (linear) and $\rho = 2$ (quadratic). Gather the Taylor series coefficients into a vector, $\boldsymbol{\beta}^{\{\rho\}}(x, y)$, defined as

$$\boldsymbol{\beta}^{\{0\}}(x,y) \equiv [z(x,y)] \tag{5}$$

$$\boldsymbol{\beta}^{\{1\}}(x,y) \equiv \begin{bmatrix} z(x,y) \\ \frac{\partial z}{\partial x}(x,y) \\ \frac{\partial z}{\partial y}(x,y) \end{bmatrix} \tag{6}$$

$$\boldsymbol{\beta}^{\{2\}}(x,y) \equiv \begin{bmatrix} z(x,y) \\ \frac{\partial z}{\partial x}(x,y) \\ \frac{\partial z}{\partial y}(x,y) \\ \frac{\partial^2 z}{\partial x^2}(x,y) \\ \frac{\partial^2 z}{\partial x \partial y}(x,y) \\ \frac{\partial^2 z}{\partial y^2}(x,y) \end{bmatrix} \tag{7}$$

for zeroth through second order fits. The length of this vector is denoted by P, with $P = 1$, 3, and 6 for $\rho = 0$, 1, and 2, respectively. The results of the analysis will be an estimate $\widehat{\boldsymbol{\beta}}_B^{\{\rho\}}(x, y)$ of this vector, dependent upon the choice of bandwidth B, and the first element of this vector is the estimated surface $\widehat{z}(x, y)$. The least squares fit is performed at a variety of different (x, y) points to obtain the estimated surface on a regular grid.

Arrange the data values and coordinates into vectors of length Q such as $\tilde{\mathbf{z}} = [\tilde{z}_1 \ \tilde{z}_2 \ \ldots \ \tilde{z}_Q]^T$, where "$T$" indicates the matrix transpose, and similarly for $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$. Now define a matrix $\mathbf{X}^{\{\rho\}}(\tilde{\mathbf{x}} - x, \tilde{\mathbf{y}} - y)$

$$\mathbf{X}^{\{0\}}(\tilde{\mathbf{x}} - x, \tilde{\mathbf{y}} - y) \equiv \left.\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}\right\} Q \text{ terms} \tag{8}$$

$$\mathbf{X}^{\{1\}}(\tilde{\mathbf{x}} - x, \tilde{\mathbf{y}} - y) \equiv \begin{bmatrix} 1 & \tilde{x}_1 - x & \tilde{y}_1 - y \\ 1 & \tilde{x}_2 - x & \tilde{y}_2 - y \\ \vdots & \vdots & \vdots \\ 1 & \tilde{x}_Q - x & \tilde{y}_Q - y \end{bmatrix} \tag{9}$$

$$\mathbf{X}^{\{2\}}(\tilde{\mathbf{x}} - x, \tilde{\mathbf{y}} - y) \equiv \begin{bmatrix} 1 & \tilde{x}_1 - x & \tilde{y}_1 - y & \frac{1}{2}(\tilde{x}_1 - x)^2 & (\tilde{x}_1 - x)(\tilde{y}_1 - y) & \frac{1}{2}(\tilde{y}_1 - y)^2 \\ 1 & \tilde{x}_2 - x & \tilde{y}_2 - y & \frac{1}{2}(\tilde{x}_2 - x)^2 & (\tilde{x}_2 - x)(\tilde{y}_2 - y) & \frac{1}{2}(\tilde{y}_2 - y)^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \tilde{x}_Q - x & \tilde{y}_Q - y & \frac{1}{2}(\tilde{x}_Q - x)^2 & (\tilde{x}_Q - x)(\tilde{y}_Q - y) & \frac{1}{2}(\tilde{y}_Q - y)^2 \end{bmatrix} \tag{10}$$

which has Q rows and P columns. This matrix contains the polynomial factors from the Taylor series expansion (1), evaluated at each data point. The weighting coefficients are likewise gathered into a diagonal matrix. With $\mathbf{r}(\mathbf{x}, \mathbf{y}, x, y) \equiv \sqrt{(\tilde{\mathbf{x}} - x)^2 + (\tilde{\mathbf{y}} - y)^2}$, form the diagonal

matrix

$$\mathbf{W}_B\left(\mathbf{r}(\mathbf{x},\mathbf{y},x,y)\right) = \begin{bmatrix} K_B(r_1) & 0 & \cdots & 0 \\ 0 & K_B(r_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & K_B(r_Q) \end{bmatrix} \tag{11}$$

which is of dimension $Q \times Q$.

The least squares estimate $\widehat{\boldsymbol{\beta}}_B^{\{\rho\}}$ of the vector $\boldsymbol{\beta}^{\{\rho\}}$ at grid point (x, y) is found by minimizing

$$\left| \mathbf{W}_B\left(\tilde{\mathbf{x}},\tilde{\mathbf{y}},x,y\right)\left[\tilde{\mathbf{z}} - \mathbf{X}^{\{\rho\}}\left(\tilde{\mathbf{x}},\tilde{\mathbf{y}},x,y\right)\widehat{\boldsymbol{\beta}}_B^{\{\rho\}}\left(x,y\right)\right]\right|^2 \tag{12}$$

and is obtained by varying the terms in the estimated vector $\widehat{\boldsymbol{\beta}}_B^{\{\rho\}}\left(x,y\right)$. The estimated surface $\widehat{z}(x,y)$ is just the first element of this vector. Note that for the $\rho = 1$ case, writing out the matrix multiplications, we find that (12) is equivalent to (4). Thus (12) is just a compact way to rewrite the least squares error in the vicinity of each grid point (x, y) for constant, linear, or quadratic fits.

Using standard least squares theory, we can immediately write the solution as

$$\boldsymbol{\beta}_B^{\{\rho\}}\left(x,y\right) = \left[\mathbf{X}^{\{\rho\}T}\left(\tilde{\mathbf{x}},\tilde{\mathbf{y}},x,y\right)\mathbf{W}_B\left(\tilde{\mathbf{x}},\tilde{\mathbf{y}},x,y\right)\mathbf{X}^{\{\rho\}}\left(\tilde{\mathbf{x}},\tilde{\mathbf{y}},x,y\right)\right]^{-1} \times \mathbf{X}^{\{\rho\}T}\left(\tilde{\mathbf{x}},\tilde{\mathbf{y}},x,y\right)\mathbf{W}_B\left(\tilde{\mathbf{x}},\tilde{\mathbf{y}},x,y\right)\tilde{\mathbf{z}} \tag{13}$$

where the superscript"-1" is the matrix inverse. This expression is perhaps more clear if we omit the arguments on the right-hand side, obtaining

$$\boldsymbol{\beta}_B^{\{\rho\}}\left(x,y\right) = \left[\mathbf{X}^{\{\rho\}T}\mathbf{W}_B\mathbf{X}^{\{\rho\}}\right]^{-1}\mathbf{X}^{\{\rho\}T}\mathbf{W}_B\,\tilde{\mathbf{z}}. \tag{14}$$

This equation may be implemented with a loop over grid points together with a numerical package for evaluating the matrix inverse. Note that the matrix which is inverted is relatively small, $P \times P$.

4 Modifications for the sphere

It is straightforward to adapt the method for scattered data on the sphere. Now the surface $z(\phi,\theta)$ is a function of both latitude ϕ and longitude θ, and is sampled at Q data points $\tilde{z}_q$ at latitudes $\tilde{\phi}_q$ and longitudes $\tilde{\theta}_q$, with $q = 1, 2, \ldots, Q$. As before, we arrange the data and the sample coordinates into vectors of length Q such as $\tilde{\mathbf{z}} = \left[\tilde{z}_1\ \tilde{z}_2\ \ldots\ \tilde{z}_Q\right]^T$ and similarly for $\widetilde{\boldsymbol{\phi}}$ and $\widetilde{\boldsymbol{\theta}}$. Based on this irregularly sampled data, we desire to obtain an estimate of $z(\phi,\theta)$ on a regular grid. For convenience, we assume for the time being that no data points are farther than $\pi/2$ radians from the grid point location (ϕ,θ).

Latitude and longitude are not appropriate to use directly in the local polynomial fit. Because the latitude/longitude coordinate system varies in its resolution over the sphere, a linear fit (for example) to these coordinates would vary in its ability to represent the local structure of a surface $z(\phi,\theta)$. Instead, it is convenient to transform the data locations $\left(\widetilde{\boldsymbol{\phi}},\widetilde{\boldsymbol{\theta}}\right)$ into a local tangent plane coordinate system about the grid point of interest.

Consider a plane tangent to the sphere at latitude ϕ and longitude θ, with the x-coordinate in this plane corresponding to east-west location, and the y-coordinate corresponding to north-south location. The (x, y) origin of the tangent plane is at the tangent point (ϕ,θ) where the plane touches the sphere. The (x, y) coordinates of the data in the tangent plane are then

$$\widetilde{\mathbf{x}}(\widetilde{\boldsymbol{\phi}},\widetilde{\boldsymbol{\theta}},\phi,\theta) = R_e\cos\widetilde{\boldsymbol{\phi}}\,\sin\left(\widetilde{\boldsymbol{\theta}}-\theta\right) \tag{15}$$

$$\widetilde{\mathbf{y}}(\widetilde{\boldsymbol{\phi}},\widetilde{\boldsymbol{\theta}},\phi,\theta) = R_e\cos\phi\sin\widetilde{\boldsymbol{\phi}} - R_e\sin\phi\cos\widetilde{\boldsymbol{\phi}}\,\cos\left(\widetilde{\boldsymbol{\theta}}-\theta\right) \tag{16}$$

where R_e is the radius of the earth. These coordinates are a function both of the data latitude $\widetilde{\boldsymbol{\phi}}$ and longitude $\widetilde{\boldsymbol{\theta}}$, as well as of the grid point location (ϕ, θ). The meaning of these equations can be visualized as follows. From every point in the tangent plane, draw a perpendicular line to the surface of the sphere. Then from location $(\widetilde{x}_q, \widetilde{y}_q)$ in the tangent plane, a perpendicular line extends to the sphere which touches the qth data point $(\widetilde{\phi}_q, \widetilde{\theta}_q)$. The tangent plane equations (15–16) are surely well known, but since we could not find a reference, a derivation is provided in Appendix A.

Once the tangent plane equations have been computed, the exact great circle distances between the data locations and the grid point (ϕ, θ) are then given via

$$\mathbf{d}(\widetilde{\boldsymbol{\phi}}, \widetilde{\boldsymbol{\theta}}, \phi, \theta) = 2R_e \arcsin\left(\frac{1}{\sqrt{2}}\sqrt{1 - \sqrt{1 - \frac{\widetilde{\mathbf{x}}^2 + \widetilde{\mathbf{y}}^2}{R_e^2}}}\right) \tag{17}$$

which is also derived in Appendix A. This is in fact equivalent to the well-known *haversin* formula for great circle distance (e.g. Wikipedia, 2009). The haversin is defined as

$$\text{haversin}(\theta) \equiv \sin^2(\theta/2) \tag{18}$$

and the haversin version of the great circle distance formula is

$$\mathbf{d}(\widetilde{\boldsymbol{\phi}}, \widetilde{\boldsymbol{\theta}}, \phi, \theta) = R_e \,\text{haversin}^{-1}\left[\text{haversin}\left(\widetilde{\boldsymbol{\phi}} - \phi\right) + \cos\theta \cos\widetilde{\boldsymbol{\theta}}\,\text{haversin}\left(\widetilde{\boldsymbol{\theta}} - \theta\right)\right]. \tag{19}$$

A few line of algebra shows that this is the same as (17). Computationally it is more efficient to compute the great circle distance from the tangent plane equations (which are needed anyway), rather than returning to the latitudes and longitudes.

An important point is that (17) assumes that the data point and the grid point are not further than $\pi/2$ radians apart. This is because data points that are further away correspond to a *second* intersection of the perpendicular line from the tangent plane to the sphere, on the far side of the sphere. While this is not a limitation since the weighting functions we will use decay to zero at far smaller distances, it is necessary to supplement (17) in order to specifically exclude such distant data points. See Appendix A for a simple algorithm for doing so.

The local polynomial fitting on the sphere can then be accomplished in a virtually identical manner as for the plane. As before, we define a matrix $\mathbf{X}^{\{\rho\}}(\widetilde{\boldsymbol{\phi}}, \widetilde{\boldsymbol{\theta}}, \phi, \theta)$ for $\rho = 0, 1,$ or 2

$$\mathbf{X}^{\{0\}}(\widetilde{\boldsymbol{\phi}}, \widetilde{\boldsymbol{\theta}}, \phi, \theta) \equiv \left.\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}\right\} Q \text{ terms} \tag{20}$$

$$\mathbf{X}^{\{1\}}(\widetilde{\boldsymbol{\phi}}, \widetilde{\boldsymbol{\theta}}, \phi, \theta) \equiv \begin{bmatrix} 1 & \widetilde{x}_1 & \widetilde{y}_1 \\ 1 & \widetilde{x}_2 & \widetilde{y}_2 \\ \vdots & \vdots & \vdots \\ 1 & \widetilde{x}_Q & \widetilde{y}_Q \end{bmatrix} \tag{21}$$

$$\mathbf{X}^{\{2\}}(\widetilde{\boldsymbol{\phi}}, \widetilde{\boldsymbol{\theta}}, \phi, \theta) \equiv \begin{bmatrix} 1 & \widetilde{x}_1 & \widetilde{y}_1 & \frac{1}{2}\widetilde{x}_1^2 & \widetilde{x}_1\widetilde{y}_1 & \frac{1}{2}\widetilde{y}_1^2 \\ 1 & \widetilde{x}_2 & \widetilde{y}_2 & \frac{1}{2}\widetilde{x}_2^2 & \widetilde{x}_2\widetilde{y}_2 & \frac{1}{2}\widetilde{y}_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \widetilde{x}_Q & \widetilde{y}_Q & \frac{1}{2}\widetilde{x}_Q^2 & \widetilde{x}_Q\widetilde{y}_Q & \frac{1}{2}\widetilde{y}_Q^2 \end{bmatrix} \tag{22}$$

which gathers together the various terms needed for the polynomial fit. The weighting matrix then becomes

$$\mathbf{W}_B(\widetilde{\boldsymbol{\phi}}, \widetilde{\boldsymbol{\theta}}, \phi, \theta) \equiv \begin{bmatrix} K_B\left(d_1(\widetilde{\phi}_1, \widetilde{\theta}_1, \phi, \theta)\right) & 0 & \cdots & 0 \\ 0 & K_B\left(d_2(\widetilde{\phi}_1, \widetilde{\theta}_1, \phi, \theta)\right) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & K_B\left(d_Q(\widetilde{\phi}_1, \widetilde{\theta}_1, \phi, \theta)\right) \end{bmatrix} \tag{23}$$

which is again of dimension $Q \times Q$. With these modifications, the solution is then simply (13)

$$\widehat{\boldsymbol{\beta}}_B^{\{\rho\}}(\phi, \theta) = \left[\mathbf{X}^{\{\rho\}T}\left(\widetilde{\boldsymbol{\phi}}, \widetilde{\boldsymbol{\theta}}, \phi, \theta\right) \mathbf{W}_B\left(\widetilde{\boldsymbol{\phi}}, \widetilde{\boldsymbol{\theta}}, \phi, \theta\right) \mathbf{X}^{\{\rho\}}\left(\widetilde{\boldsymbol{\phi}}, \widetilde{\boldsymbol{\theta}}, \phi, \theta\right)\right]^{-1} \times \mathbf{X}^{\{\rho\}T}\left(\widetilde{\boldsymbol{\phi}}, \widetilde{\boldsymbol{\theta}}, \phi, \theta\right) \mathbf{W}_B\left(\widetilde{\boldsymbol{\phi}}, \widetilde{\boldsymbol{\theta}}, \phi, \theta\right) \tilde{\mathbf{z}} \tag{24}$$

where the terms in the vector $\widehat{\boldsymbol{\beta}}_B^{\{\rho\}}(\phi, \theta)$ are now interpreted as the derivatives in a local tangent plane expansion about the point (ϕ, θ). The first term in the vector $\widehat{\boldsymbol{\beta}}_B^{\{\rho\}}(\phi, \theta)$ remains the estimated surface $\widehat{z}(\phi, \theta)$.

5 Algorithm Status

We have implemented local polynomial fitting on the plane and on the sphere as a part of a freely available Matlab toolbox for use by the community. It is numerically optimized by direct (i.e., loopless) calculation of the matrix inverses. The spherical algorithm additional has enhancements to minimize unnecessary distance computations between distant points, and to avoid the exchange of information between ocean basins (e.g. across the isthmus separating the Pacific Ocean from the Caribbean Sea). A second mode of operation, called "nearest-neighbor" mode, is also implemented; this permits the bandwidth to be *locally* chosen to encompass the nearest say N data points.

Local polynomial fitting has a number of useful extensions which will be considered in the future. For first-order and higher-order fits, *gradients* of the surface are estimated in addition to the surface itself. This could be useful for Aquarius as the sea surface salinity gradient is a quantity of interest. Estimation of errors, both *bias* due to model mismatch and *variance* due to random noise, are possible (Ruppert and Wand, 1994; Fan and Gijbels, 1996) and will be implemented in the future. Other possibilities included automated methods for choosing fit radius and/or polynomial order, alternative choices of weighting functions, and anisotropic weighting.

The main remaining task is to determine what bandwidth B and polynomial order ρ are most appropriate for Aquarius. To a large extent, the bandwidth B is fixed on account of the 150 km decorrelation scale in the mission requirements. The most appropriate choices of these two parameters depends upon the magnitude and nature of the random variability present in the estimated salinity values. Determination of the optimal values can be accomplished with a detailed investigation of output from the Aquarius simulator experiments.

6 Examples

We now present several preliminary showing a large reduction in mapping errors for the first order fit compared with the zeroth order fit, and for the second order fit compared to the first-order fit. A main advantage of the higher-order fits is the minimization of artifacts due to the irregular spacing of the Aquarius sampling on account of the tri-beam configuration.

More to do here...

A Spherical Details

The tangent plane equations (15–16) are not necessarily obvious, so a derivation is provided here. Let x', y', and z' be a Cartesian coordinate system with the origin at the center of the sphere, and let x and y be the coordinates in a tangent plane which touches the sphere at (ϕ_o, θ_o). The coordinate x is positive eastward at the tangent point, while y is positive northward; the x' axis is chosen to be parallel to the x axis. We aim to find the x and y coordinates of some other point (ϕ, θ), whose angular distance to (ϕ_o, θ_o) is assumed to be less than $\pi/2$ radians.

The x' and z' coordinates of the point (ϕ, θ) can be written down at once as

$$\begin{aligned} x' &= R_e \cos\phi \sin(\theta - \theta_o) && (25)\\ y' &= -R_e \cos\phi \cos(\theta - \theta_o) && (26)\\ z' &= R_e \sin\phi. && (27) \end{aligned}$$

Now let $h(x, y)$ be the perpendicular distance from the tangent plane to the sphere, with $h(0, 0) = 0$ by definition. In terms of x and y, x' and z' are then

$$\begin{aligned} x' &= x && (28)\\ z' &= [R_e - h(x, y)] \sin\phi_o + y \cos\phi_o && (29) \end{aligned}$$

and we see that x and x' are identical. Thus we obtain

$$y = R_e \left[\frac{\sin\phi}{\cos\phi_o} - \tan\phi_o \right] + h(x, y) \tan\phi_o \qquad (30)$$

as an equation for y. One other other hand, $h(x, y)$ is found from the Pythagorean theorem

$$[R_e - h(x, y)]^2 + \left(x^2 + y^2\right) = R_e^2 \qquad (31)$$

where we consider a triangle whose hypotenuse extends from the center of the earth to the point ϕ, θ. This leads to

$$h(x, y) = R_e \left[1 - \sqrt{1 - \frac{x^2 + y^2}{R_e^2}} \right] \qquad (32)$$

as an equation for $h(x, y)$. Inserting this into the square of (30), and with a bit of algebra involving trigonometric identifies, (16) follows. Note that the reference point is denoted by (ϕ, θ) in the text, but by (ϕ_o, θ_o) in this appendix.

It is necessary to identify those pairs of points which are separated by more that $\pi/2$ radians. To avoid excessive computation, this should be done without computing the full spherical distance. Instead simply note that the coordinates of the reference point, with respect to the three-dimensional Cartesian coordinate system, are

$$\begin{aligned} x'_o &= 0 && (33)\\ y'_o &= -R_e \cos\phi_o && (34)\\ z'_o &= R_e \sin\phi_o && (35) \end{aligned}$$

with x'_o vanishing by definition. Thus when $y'_o y' + z'_o z' > 0$, the dot product between the vectors representing these two points is positive, and therefore they are separated by no more than $\pi/2$ radians.

Derivation of the distance formula (17) is straightforward. The great circle distance between the point at (ϕ_o, θ_o) and the point at (ϕ, θ) is $d = 2R_e\Delta\Phi$, where $\Delta\Phi$ is half the angular distance between the two points. This is given by

$$\Delta\Phi = \arcsin\left(\frac{1}{2}\frac{\sqrt{x^2+y^2+h^2(x,y)}}{R_e}\right) = \arcsin\left(\frac{1}{\sqrt{2}}\sqrt{1-\frac{h(x,y)}{R_e}}\right) \tag{36}$$

where we have noted that

$$x^2+y^2+h^2(x,y) = x^2+y^2+2R_e^2-2R_eh(x,y)-x^2-y^2 = 2R_e^2-2R_eh(x,y) \tag{37}$$

on account of (32). Equation (17) follows at once.

References

Fan, J. and Gijbels, I.: Local polynomial modelling and its applications, Chapman and Hall, 1996.

Gaspar, P., Labroue, S., Ogor, F., Lafitte, G., Marchal, L., and Rafanel, M.: Improving nonparametric estimates of the sea state bias in radar altimeter measurements of sea level, J. Atmos. Ocean Tech., 19, 1690–1707, 2002.

Ruppert, D. and Wand, M. P.: Multivariate locally weighted least squares regression, Ann. Stat., 22, 1346–1370, doi:10.1214/aos/1176325632, 1994.

Weisstein, E. W.: Great Circle, `http://mathworld.wolfram.com/GreatCircle.html`, 2008.

Wikipedia: Haversine formula — Wikipedia, The Free Encyclopedia, `http://en.wikipedia.org/w/index.php?title=Haversine_formula&oldid=272334912`, online; accessed 24-February-2009, 2009.